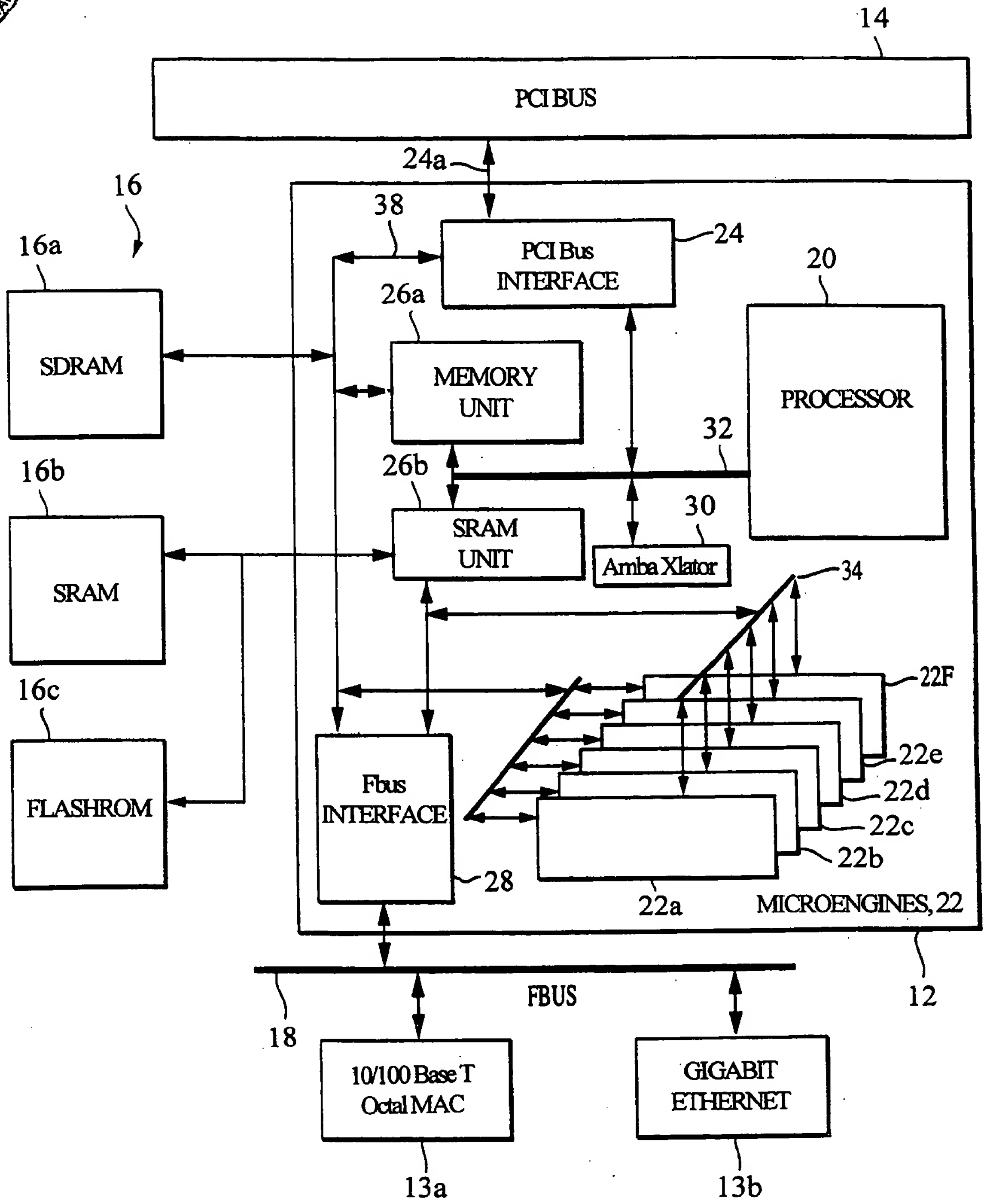

14
PCI BUS
24a
16
38
PCI Bus INTERFACE
24
20
16a
26a
SDRAM
MEMORY UNIT
PROCESSOR
32
26b
16b
30
SRAM UNIT
Amba Xlator
SRAM
34
22F
16c
22e
22d
FLASHROM
Fbus INTERFACE
22c
22b
28
22a
MICROENGINES, 22
FBUS
12
18
10/100 Base T Octal MAC
GIGABIT ETHERNET
13a
13b

FIG. 1

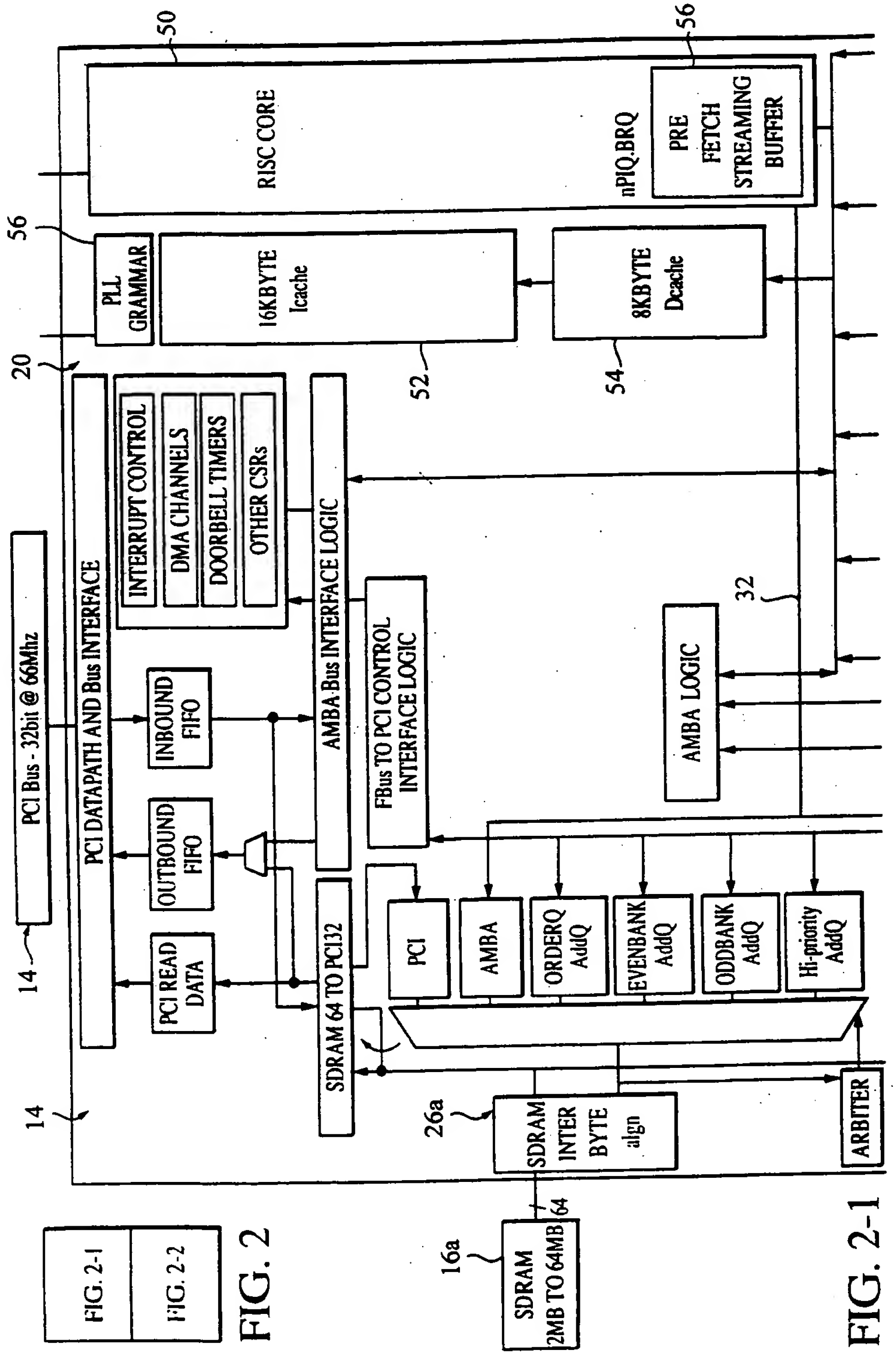

50
56
RISC CORE
nPIQ.BRQ
PRE FETCH STREAMING BUFFER
56
PLL GRAMMAR
16KBYTE Icache
8KBYTE Dcache
20
52
54
PCI Bus - 32bit @ 66Mhz
PCI DATAPATH AND Bus INTERFACE
INTERRUPT CONTROL
DMA CHANNELS
DOORBELL TIMERS
OTHER CSRs
AMBA Bus INTERFACE LOGIC
INBOUND FIFO
OUTBOUND FIFO
PCI READ DATA
FBus TO PCI CONTROL INTERFACE LOGIC
32
AMBA LOGIC
SDRAM 64 TO PCI32
PCI
AMBA
ORDERQ AddQ
EVENBANK AddQ
ODDBANK AddQ
Hi-priority AddQ
14
14
26a
SDRAM INTER BYTE algn
ARBITER
64
16a
SDRAM 2MB TO 64MB
FIG. 2-1
FIG. 2-2

FIG. 2

FIG. 2-1

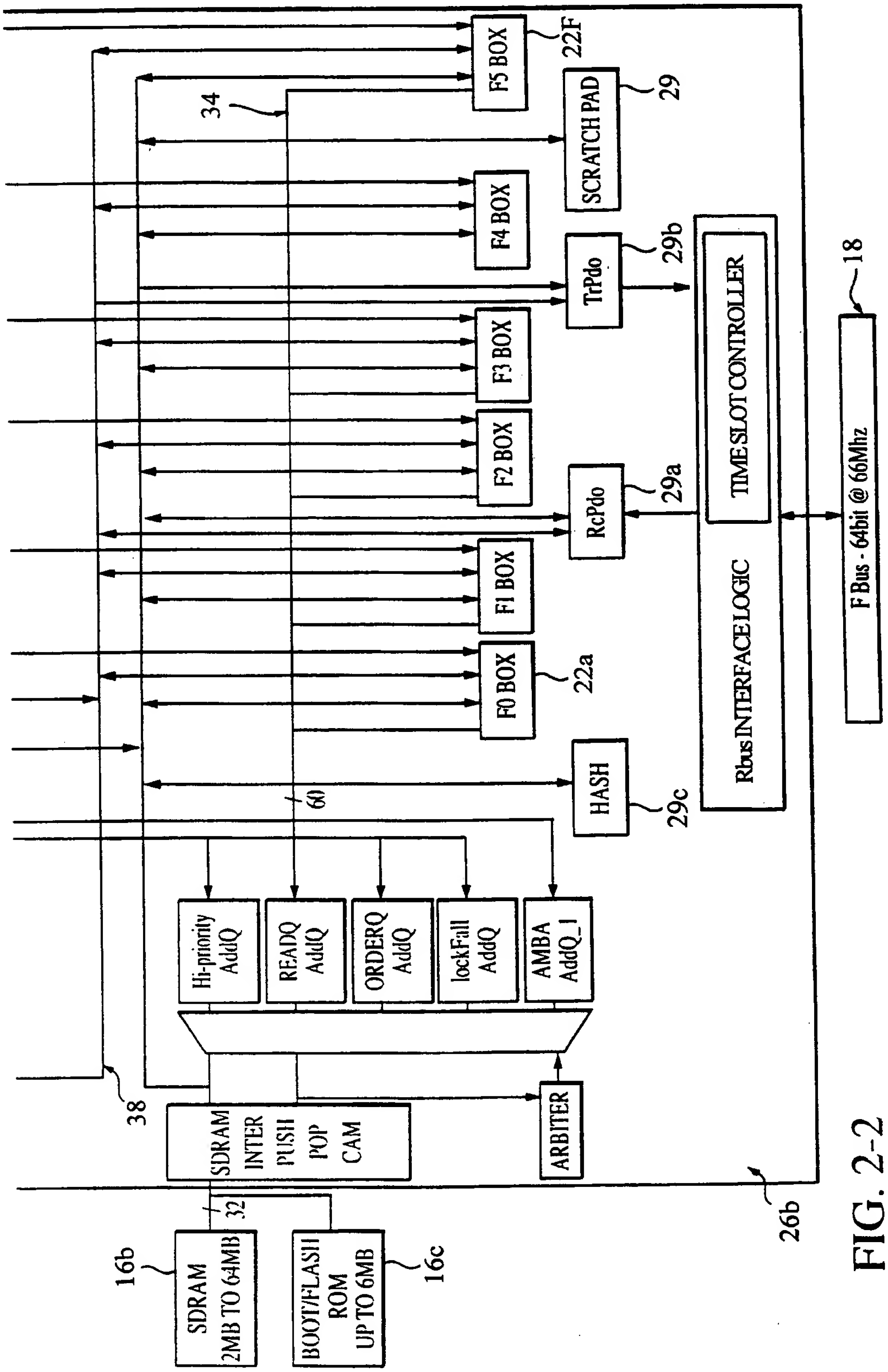

F5 BOX
22F
SCRATCH PAD
29
34
F4 BOX
TrPdo
29b
TIME SLOT CONTROLLER
18
F3 BOX
F2 BOX
RcPdo
29a
F Bus - 64bit @ 66Mhz
F1 BOX
F0 BOX
22a
Rbus INTERFACE LOGIC
HASH
29c
60
Hi-priority AddQ
READQ AddQ
ORDERQ AddQ
lockFail AddQ
AMBA AddQ_1
38
SDRAM INTER PUSH POP CAM
ARBITER
26b
32
16b
SDRAM 2MB TO 64MB
BOOT/FLASH ROM UP TO 6MB
16c

FIG. 2-2

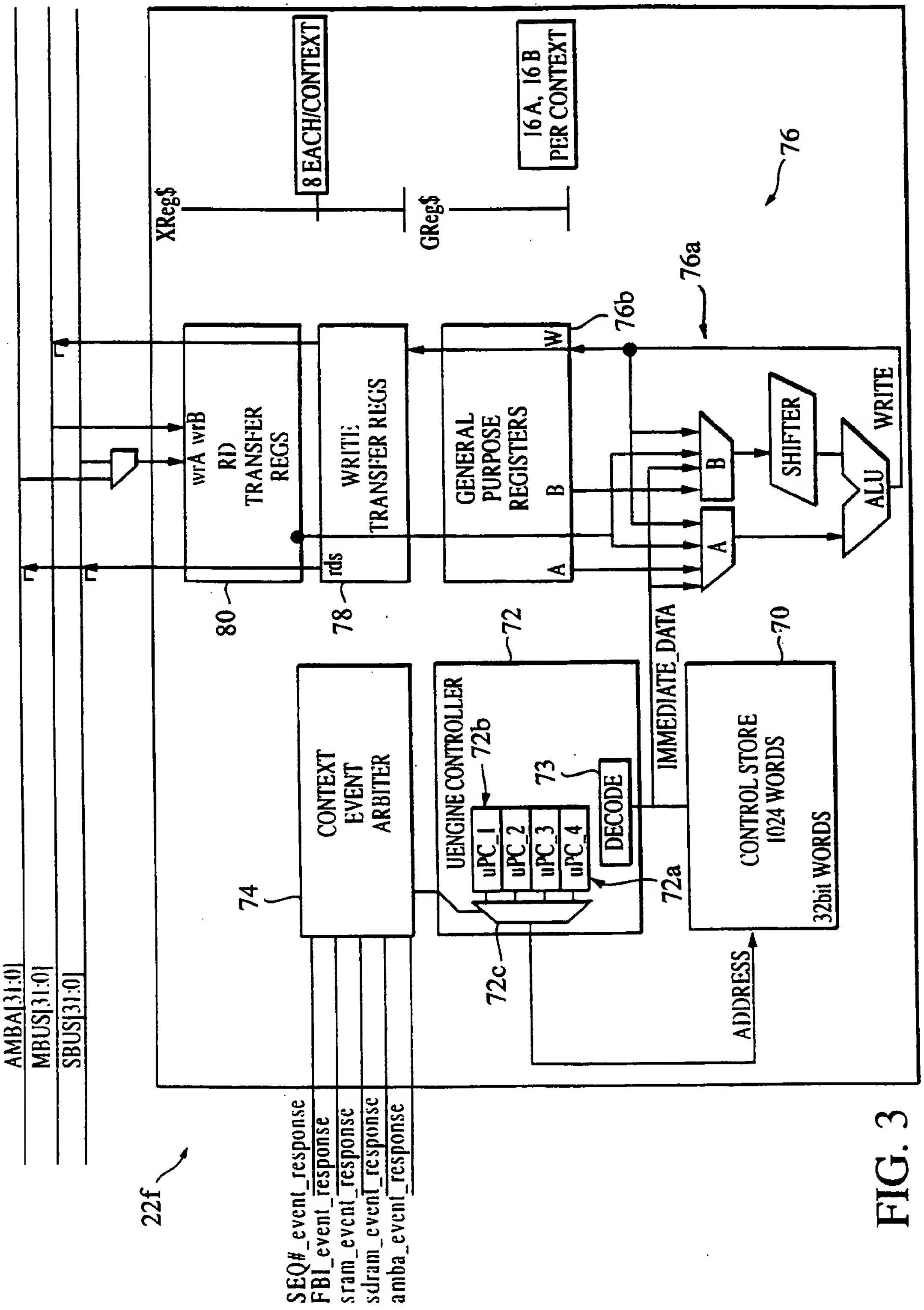

AMBA[31:0]
MBUS[31:0]
SBUS[31:0]
XReg$
8 EACH/CONTEXT
GReg$
16 A, 16 B PER CONTEXT
76
76a
76b
W
wrA wrB RD TRANSFER REGS
WRITE TRANSFER REGS
rds
GENERAL PURPOSE REGISTERS
B
A
SHIFTER
ALU
WRITE
80
78
72
70
IMMEDIATE_DATA
CONTEXT EVENT ARBITER
UENGINE CONTROLLER
72b
73
uPC_1
uPC_2
uPC_3
uPC_4
DECODE
72a
CONTROL STORE 1024 WORDS
32bit WORDS
74
72c
ADDRESS
22f
SEQ#_event_response
FBI_event_response
sram_event_response
sdram_event_response
amba_event_response

FIG. 3

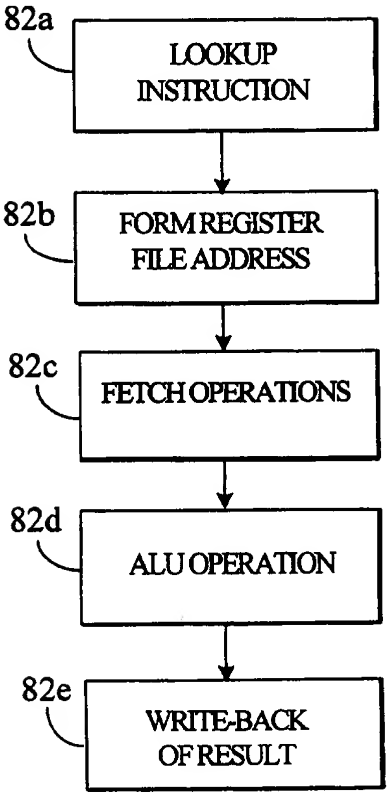
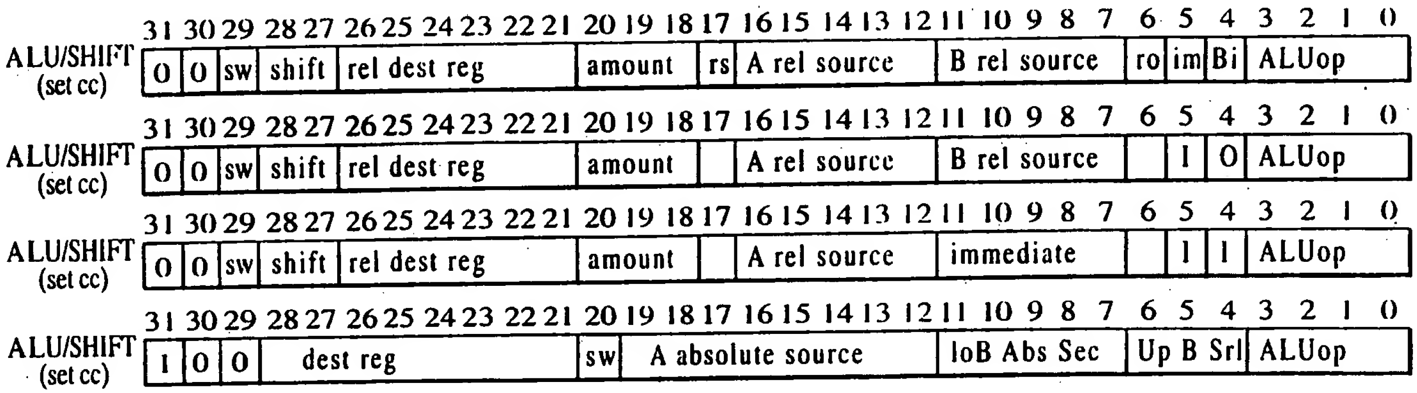

82a
LOOKUP INSTRUCTION
82b
FORM REGISTER FILE ADDRESS
82c
FETCH OPERATIONS
82d
ALU OPERATION
82e
WRITE-BACK OF RESULT

FIG. 4



Shift Decode:

(rs,r0) decode ([31:0] shifts into [63:32] and take [63;32]):

00 = left rotate
01 = right shift (32-ShfAmt = Right Shift Amt)
10 = left shift
11= double shift (upper A-op shifts into lower B-op)

===> "left rotate" of zero gives zero shift (therwise zero amount signifies indirect shift)

ALU-OP decode:

0000 = B
0001 = ~B
0010 = A&B (and)
0011= A&~B (and~)

0100 = ~A&B (~and)
0101 =XOR
0110 = OR
0111= mul-stuff

1000 = A-B
1001 = B-A
1010 =
1011=

1100 = A+B(8)
1101 = A+B(16)
1110 = A+B
1111= A+B+Cin

FIG. 5